JERICHO SYSTEMS WHITE PAPER:
Key Decision Points for Using Open Source Software in Access Control Solutions

January 2014

CONTENTS

PURPOSE

This white paper discusses the use of open source software within access control solutions and presents key decision points to be considered when adopting open source products within enterprise-level access control.

This discussion relates to the following topics:

- Supply Chain Risk Management (SCRM)
- Information Assurance (IA) Products
- Software Development Lifecycle Management (SDLC)
- Component Lifecycle Management (CLM)
- Software Security Development Lifecycle (SDL)
- Attribute Based Access Control (ABAC)
- Open protocols, including Security Assertion Markup Language (SAML) and eXtensible Access Control Markup Language (XACML)

BACKGROUND

Information Technology (IT) departments within the U.S. Department of Defense (DoD) and other large enterprises are tasked with protecting sensitive data resources while simultaneously enabling authorized users to collaborate in real-time. Traditionally, the technologies required to accomplish this task would be either developed in-house or acquired from industry developers. Today the majority of IT solutions – whether developed internally or acquired externally – incorporate open source concepts and components, including *open architecture*, *open protocols*, and *open source software*.

An open *architecture* is a design philosophy that incorporates specifications that are published publicly and are usually supported by a community of experts that collaboratively created the specifications and will maintain and update them. Examples of such open architecture organizations include the Internet Engineering Task Force (IETF),[1] World Wide Web Consortium (W3C),[2] and Organization for the Advancement of Structured Information Standards (OASIS).[3] Many large companies, including Oracle, IBM, and HP, embrace the open architecture philosophy and donate back to the community in time and resources.

---

[1] Internet Engineering Task Force, http://www.ietf.org/

[2] World Wide Web Consortium, http://www.w3.org/Consortium/

[3] Organization for the Advancement of Structured Information Standards, https://www.oasis-open.org/

Open *protocols* are technical guidelines that offer a standardized framework for sharing information within IT solutions and help foster software interoperability. For example, SAML[4] and XACML[5] are open protocols that describe how information should be transmitted between components of an access control solution.

Open source *software* products, by contrast, are building blocks of code with varying copyright and licensing agreements that developers can use, modify, and reuse. Open source software can also be found in Government-Off-The-Shelf (GOTS) and Commercial-Off-The-Shelf (COTS) products.

Open source concepts and components can offer benefits. Many large companies, including Oracle, IBM, and HP, embrace open architectures and open protocols because they enable interoperability. Government and industry organizations are drawn to open source software by the promise of reduced cost and development burden.

As a provider of commercial access control solutions to the DoD and other enterprises, Jericho Systems Corporation incorporates "open" principles – including open architecture, open protocols, and open source software – into solutions built with the EnterSpace® Decisioning Service (ESDS)[6] policy evaluation engine. Our *open-friendly* solutions deliver state-of-the-art ABAC technology and compete with other GOTS and COTS solutions, such as Oracle Entitlements Server (OES);[7] solutions; and open source products.

This white paper explains our best practice recommendations for making cost-effective, risk-aware decisions when evaluating access control solutions, with special attention to the unique needs of the DoD and other federal agencies. In addition, this discussion illustrates how Jericho Systems adds value to your access control effort.

THE OPEN SOURCE DEBATE

Increasingly, access control solutions incorporate *open architecture, open protocols,* and *open source software* to enable authentication and authorization capabilities to permit and deny access to sensitive resources. Recently, Oracle published a white paper challenging the use of open source software by the DoD.[8] This white paper attracted news coverage[9] and,

---

[4] OASIS Security Services Technical Committee, https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security

[5] OASIS eXtensible Access Control Markup Language Technical Committee, https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml

[6] EnterSpace® Decisioning Service (ESDS) from Jericho Systems Corporation, http://www.jerichosystems.com/products/decisioning.html

[7] Oracle Entitlements Server, http://www.oracle.com/technetwork/middleware/oes/overview/index.html

[8] The Department of Defense (DoD) and Open Source Software, Oracle White Paper, September 2013, http://www.oracle.com/us/products/middleware/cloud-app-foundation/weblogic/dod-and-open-source-software-2012277.pdf

unsurprisingly, the company's recommendation against using open source software solutions was criticized within the software developer community.[10]

Opinions for and against the use of open source software have been around for a long time. Prior to Oracle's white paper, another recent debate centered around a statement published by Cryptocat on their blog:

> *"We will commit failures dozens, if not hundreds of times more in the coming years, and we only ask you to be vigilant and careful. This is the process of open source security."*[11]

The open source versus proprietary software debate typically elicits a range of concerns. For example, a CSO article about the Cryptocat vulnerability presented these worries:[12]

> *"… because open-source software is developed by an unpaid group of engineers, there are going to be security lapses"*
>
> *"Since open source software isn't owned by anyone, there are no dedicated software maintenance people and enhancements are made by whoever can and wants them"*
>
> *"… developers paid to build software have more at stake in getting it right"*
>
> *"commercial developers depend on the quality of their product to pay their mortgages and feed their families… this forces commercial developers to pay more attention to bugs and to do more rigorous testing"*
>
> *"…companies can be held liable for software left insecure due to negligence"*
>
> *"…failures of Cryptocat are not failures of open-source versus closed-source development, but rather a failure in the secure development process"*

While Jericho Systems Corporation is a proponent of open architectures, open protocols, and open source software, we recognize the need for caution. Mission-critical enterprises need more than a cookie-cutter, "always-do-it-this-way" approach to IT development, integration, and procurement. Choosing an access control solution is a complex decision. By using the key decision points identified in this paper, we believe it's possible to find the "sweet spot" of IT decision-making that combines the best of open and commercial development.

---

[9] Oracle says open source has no place in military apps: Unless it's open source from Oracle, of course,15 October 2013, http://www.theregister.co.uk/2013/10/15/oracle_says_open_source_has_no_place_in_military_apps/

[10] Oracle Attacks Open Source; Says Community-Developed Code Is Inferior, 15 October 2013, http://developers.slashdot.org/story/13/10/15/1828211/oracle-attacks-open-source-says-community-developed-code-is-inferior?sdsrc=next

[11] New Critical Vulnerability in Cryptocat: Details, 04 July 2013, https://blog.crypto.cat/2013/07/new-critical-vulnerability-in-cryptocat-details/

[12] Cryptocat vulnerability excuse sparks debate over open source, 09 July 2013, http://www.csoonline.com/article/736053/cryptocat-vulnerability-excuse-sparks-debate-over-open-source

KEY DECISION POINTS FOR SCRM

The practice of *Supply Chain Risk Management* (SCRM) recognizes that the manner in which IT products are sourced has security, financial, and other implications for the enterprise. Access control solutions, in particular, deserve careful scrutiny. Each component needs to be carefully chosen and rigorously controlled or it will pose undue risk for sensitive data and applications. Due diligence typically falls to the solution developers, who must implement strict controls and provide documentation to assure their customers (including designated approving authorities, authorizing officials, integrators, etc.) that each component in the software solution is trustworthy.

Whether the access control solution is an open source software (OSS), GOTS, or COTS product, a sensitive or classified network must be assured that all source code has been tested for vulnerabilities and that customers will receive timely fixes if/when vulnerabilities are discovered. Making that determination requires a deep understanding of each software component in the solution, where it comes from, and a secure software development life cycle (SDL/SDLC) process that supports continuous research and testing to find and fix vulnerabilities. Vigilance and continuous research and testing of open source software are especially important given the ability of malicious code writers to analyze open source code and craft malicious logic for insertion into later versions.

The balance of this section presents four key decision points to guide your evaluation of open source within the context of access control.

#1: PROVENANCE OF SOFTWARE AND COMPONENTS

When choosing to acquire or deploy an access control solution that incorporates open source software, we recommend considering its provenance, or history. For example, it is advisable to scrutinize software originating in "grey market" channels such as:[13]

- online reuse repositories
- open source repositories
- freeware and shareware sites
- individuals' websites

Software originating in such channels may incorporate not only open source applications, but open source components, such as libraries, which can introduce risks. In fact, the Open Web Application Security Project (OWASP) included "Using Components with Known

[13] Supply Chain Risk Management and the Software Supply Chain, https://www.owasp.org/images/7/77/BoozAllen-AppSecDC2010-sw_scrm.pdf

Vulnerabilities"[14] on its 2013 Top 10 List of the most important web application security weaknesses.[15]

The odds of using components with known vulnerabilities are surprisingly high, according to a recent Dark Reading article:[16]

> *"According to a study last year by Aspect, more than one in four common libraries used for Java, among a pool of 113 million libraries downloads, were used with known vulnerabilities…*
>
> *… if you have 100 libraries in your app and there's a one in four chance you're going to be downloading one with a known vulnerability, the chances of you having at least one library with a known vulnerability is pretty darned high*
>
> *White Source found that of among its new customers, 85 percent of applications loaded to its Open Source Lifecycle Management contained at least some out-of-date components in their code base.*
>
> *Another study by component life cycle management company Sonotype points to some of the root causes of the vulnerabilities. Among a sample size of 3500, 76 percent reported their organizations have no control over what components are being used in software development projects, and 65 percent don't have an inventory of components used in their projects."*

A recent article in CrossTalk magazine[17] agrees that open source components can become a Pandora's box:

> *"Whether provided by commercial vendors or open source initiatives… components can introduce significant management, security and licensing challenges.*
>
> *[An application] may contain hundreds or thousands of externally sourced components from dozens or hundreds of original suppliers. Each of these components has its own lifecycle, its own bug fixes and feature enhancements, and its own potential risks.*
>
> *[A] single flawed component could cause significant problems for the user. In the worst case, these problems could lead to security breaches, data leaks, stability, and performance issues, or legal actions related to intellectual property."*

---

[14] Open Web Application Security Project: Using Components with Known Vulnerabilities https://www.owasp.org/index.php/Top_10_2013-A9-Using_Components_with_Known_Vulnerabilities

[15] Open Web Application Security Project 2013 Top 10, https://www.owasp.org/index.php/Top_10_2013-Top_10

[16] Controlling The Risks Of Vulnerable Application Libraries, 22 May 2013, http://www.darkreading.com/applications/controlling-the-risks-of-vulnerable-appl/240155396

[17] Open Source and the Software Supply Chain: A Look at Risks vs. Rewards, http://www.crosstalkonline.org/storage/issue-archives/2013/201303/201303-Jackson.pdf

This issue becomes magnified in the area of security, as underscored by an example concerning the Sonatype Central Repository[18] which is a significant source of open source components for industries around the world:

> *"Recent analysis by Aspect Security, using data from the Central Repository, uncovered widespread security vulnerabilities among the most commonly used open source components."*

Further, this article cites a Sontatype survey that found only 32% of all organizations maintain a complete list of all open source components, with their dependencies, used in production applications.

As a COTS vendor, Jericho Systems Corporation adds value for its customers by proactively managing the use of components throughout the supply chain of our development process and maintenance support. We maintain lists of every third-party software component used in our products. For example, Jericho Systems Corporation incorporates an HSQL[19] database to provide embedded policy store functionality within the EnterSpace Decisioning Service product. Jericho Systems conscientiously researches and tests the HSQL database for vulnerabilities[20] as part of the SDL/SDLC product development process and maintenance support to our customers.

#2: TESTING AND ACCREDITATION

Because of their importance to the security of an enterprise's networks and information, access control products require extra scrutiny for any vulnerability and associated risk. For national security systems,[21] access control solutions were traditionally developed and produced by the National Security Agency (NSA) and known as Government-Off-the Shelf (GOTS) products. Since 2000, Commercial-Off-The-Shelf (COTS) solutions could be acquired for national security systems and procured under the Information Assurance products category.[22]

Historically, IA products for national security systems are held to a specialized standard. They have been subject to validation and compliance requirements of National Security Telecommunications and Information Systems Security Policy (NSTISSP) #11 Common

---

[18] The Central Repository, http://www.sonatype.org/central

[19] hsqldb BSD License, http://hsqldb.org/web/hsqlLicense.html

[20] For example, searching the National Vulnerabilities Database for hsqldb: http://web.nvd.nist.gov/view/vuln/search-results?query=hsqldb&search_type=all&cves=on

[21] Committee on National Security Systems: FAQ, https://www.cnss.gov/CNSS/about/faq.cfm

[22] NIST SP 800-23, August 2000, Guidelines to Federal Organizations on Security Assurance and Acquisition/Use of Tested/Evaluated Products, http://csrc.nist.gov/publications/nistpubs/800-23/sp800-23.pdf

Criteria[23] and more recently to Committee on National Security Systems (CNSS) Policy No. 11, National Policy Governing the Acquisition of Information Assurance (IA) and IA-Enabled Information Technology Products.[24]

Federal agencies can benefit from considering the following questions before procuring an access control solution, which is considered an IA product on national security systems:

- Does the access control solution meet the CNSS Policy No. 11 via National Information Assurance Partnership/Common Criteria Evaluation and Validation Scheme (NIAP/CCEVS) protection profile certification?
- Is the access control solution listed in a Cyber Security and Information Systems Information Analysis Center (CSIAC)[25] Product Evaluation or on a DoD Approved Product List?
- Is a NSA Security Configuration Guide available and current for the access control solution[26]?
- Is a Defense Information System Agency (DISA) Security Technical Implementation Guide (STIG)/Security Content Automation Protocol (SCAP) compliant tool available and current for the access control solution?
- Does the access control solution already have a Security Control Assessment (Certification) and Authorization to Operate (Accreditation) with Reciprocity IAW CJCSI 6211.02D[27]?

Jericho Systems Corporation actively works to stay current on government requirements and guidance. We consider independent security testing and certification an important part of establishing the trustworthiness of security products.

#3: LIFE CYCLE PROCESSES

Securing an enterprise against tainted or outdated open source IT products is not a once-and-done activity. To be successful, it requires on-going iterative and multi-faceted efforts.

---

[23] National Information Assurance Partnership/Common Criteria Evaluation and Validation Scheme, https://www.niap-ccevs.org/index.cfm

[24] CNSS Policy No. 11, 10 June 2013, National Policy Governing the Acquisition of Information Assurance (IA) and IA-Enabled Information Technology Products, https://www.cnss.gov/Assets/pdf/CNSSP-11.pdf

[25] Cyber Security and Information Systems Information Analysis Center https://www.thecsiac.com/about/about-the-csiac

[26] NSA Security Configuration Guides, http://www.nsa.gov/ia/mitigation_guidance/security_configuration_guides/

[27] CJCSI 6211.02D, Defense Information Systems Network (DISN) Responsibilities, 24 January 2012, http://www.dtic.mil/cjcs_directives/cdata/unlimit/6211_02.pdf

The Software & Supply Chain Assurance chart from US CERT offers helpful high-level guidance, recommending that an organization:[28]

- Integrate supply chain knowledge into secure solutions concepts
- Incorporate SCRM into acquisition requirements
- Evaluate proposals for SCRM capabilities
- Incorporate threat assessments and evaluate capability to mitigate residual risks
- Incorporate SCRM measures into overall ICT security

Similarly, the U.S. Department of Homeland Security[29] encourages incorporating SCRM into all aspects of the systems development life cycle. Approaching SCRM at an enterprise level, or "systems-of-systems" orientation makes it possible to integrate SCRM into all relevant activities. They suggest identifying baseline pre-conditions for SCRM and functional groups within the enterprise that own SCRM enhancements. They also recommend following up by performing audits and analyses that ultimately help minimize malware, vulnerabilities, and exploitable application weaknesses.

Before choosing or using access control solutions that include open source, we recommend that your organization consider the following questions:

- Is the open source code secure and trustworthy?
- Is documentation being maintained to be current and complete; providing initial configuration, user guides, and maintenance manuals for each of the access control solution product components?
- Does your organization have subject matter experts available who are capable of identifying and rectifying back-end risks related to the solution?
- Does your organization have the time and personnel to (re)certify open source software solutions, if required?
- Are there any additional hidden costs of adopting this software or does the open source solution have examples of demonstrated Total Cost of Ownership (TCO)?

---

[28] Software & Supply Chain Assurance, US CERT, January 2013, https://buildsecurityin.us-cert.gov/sites/default/files/publications/SUMMARY_OF_5-KDPs_for_SwA%20and%20SCRM.PPTX

[29] "Supply Chain Risk & Management: Enabling Transparency for Informing & Decision Making in Reducing Residual Risk & Exposures" from U.S. Department of Homeland Security (DHS) Supply Chain Risk Management (SCRM) and Software Assurance (SwA) (Software SCRM) Program Offices, http://csrc.nist.gov/scrm/documents/workshop_oct2012/jarzombek_ict_supply_chain_workshop_oct-15-2012.pdf

- Who is in charge of architecting and configuring the access control solution in your environment(s) (e.g. selecting the types of ciphers used)?
- Do other systems depend on the trustworthiness of your systems?
- Who certifies and maintains the trustworthiness of the code in the context of your larger (enterprise) dynamic access control management solution?
- What is the process by which components of the solution will be maintained and kept current in terms of licensing, interoperability/compatibility, and standards?

We also recommend asking application developers whether they implement Component Lifecycle Management (CLM) - the practice of proactively managing components throughout the supply chain. In his April 2013 CrossTalk article,[30] Wayne Jackson describes four critical CLM steps, summarized here:

1. Inventory and audit your current component usage and identify dependencies
2. Closely analyze applications and components to identify vulnerabilities
3. Establish policies and controls to address, prevent, and exclude viral licenses, vulnerable components, and flawed components
4. Stay current: keep steps 1 and 2 up-to-date

Jericho Systems actively implements this four-step approach to CLM, tracking more than 50 third-party libraries and application programming interfaces (APIs) for just one of our products. Each component is carefully chosen based on research and testing for provenance, vulnerabilities, and support from the community such as demonstrated activity and security expertise.

#4: TOTAL COST OF OWNERSHIP

Total Cost of Ownership (TCO) can be a hotly debated topic when comparing GOTS, COTS, and OSS, but we believe it is a critically important aspect of software evaluation.

Section 4.1.3.1 of the Defense Acquisition Guidebook[31] rightly points out that a single IT system often incorporates multiple software applications with varying licensing rights and agreements. One application could have low up-front costs followed by heavy maintenance expenses, while another could incur high costs in the short-term but sustain minimal maintenance charges in the long-term. Therefore, the Guidebook recommends:

---

[30] Open Source and the Software Supply Chain: A Look at Risks vs. Rewards, http://www.crosstalkonline.org/storage/issue-archives/2013/201303/201303-Jackson.pdf

[31] Defense Acquisition Guidebook: System Engineering: Software, 23 May 2013, https://acc.dau.mil/CommunityBrowser.aspx?id=638301

- Evaluating "differences regarding acquisition and sustainment costs, performance, and the consequences on change control and sustainment of deployed systems."
- Considering concept of operations (CONOPS), maintenance plans, user audience, and level of user training "to effectively balance the cost, scheduling and potential risks in maintenance, training, and documentation"
- Evaluating acquisition and sustainment costs "using relevant metrics (size, complexity, productivity factors, quality, development organization's past performance/productivity, etc.)."

The Federal Financial Institutions Examination Council concurs. In their report, *Risk Management of Free and Open Source Software*, the council urges institutions to consider both direct and indirect costs of Free Open Source Software (FOSS).[32] This calculation can favor commercial software (emphasis ours):

> *'One of the features attracting institutions to FOSS is its complimentary or low cost for licensing and maintenance. However, the* ***indirect costs of FOSS may be higher*** *than those associated with proprietary software if existing staff requires more training than would otherwise be necessary with a proprietary product. In addition,* ***change management costs may be higher in a FOSS environment*** *if the institution implements products lacking third-party vendor support. The institution generally will bear more responsibility and spend more resources identifying, selecting, analyzing, and installing upgrades and patches.* ***Depending on the FOSS selected, other indirect costs may appear,*** *such as code reviews, documentation, and contingency planning."*

A DoD report goes even further, concluding that commercial software costs less than open source products over the long term. A task force [33] observed that the cost to maintain a line of code remains constant. As the size of software applications balloon, commercial vendors in effect provide a bulk discount against the per-line-of-code cost:

> *"[the]* ***annual cost of maintaining the Department's software-enabled capabilities could not only rise exponentially but, where the capability is enabled by open-source software, could increase by ten times the cost of similar capability*** *provided by the established and structured commercial software industry.*
>
> *This conclusion assumes that the cost of maintaining a single line of code is relatively constant over time and the maintenance cost (per SLOC) is the same for both commercial off-the-shelf*

[32] Risk Management of Free and Open Source Software, http://www.federalreserve.gov/boarddocs/srletters/2004/SR0417a1.pdf

[33] March 2009 report from The Task Force on Department of Defense Policies and Procedures for the Acquisition of Information Technology, Office of the Under Secretary of Defense for Acquisition, Technology, and Logistics http://www.acq.osd.mil/dsb/reports/ADA498375.pdf

and open source software. Clearly, the Department will have to develop a strategy to control this growth in a reasonable and practical way. That the majority of commercial code, such as for example Microsoft Windows, has grown exponentially while the cost has been nearly constant and has not tracked the lines-of-code metric, gives an even more compelling reason for DOD to develop standards and processes to use and acquire as much commercial-based code as possible."

Before selecting an access control system, we recommend you perform a close examination of the direct and indirect costs of all the necessary components. One effective tool for analysis is the downloadable *TCO Matrix created for OpenForum Europe by Deloitte* that can be found in Appendix E of Total cost of ownership of open source software: a report for the UK Cabinet Office supported by OpenForum Europe.[34]

CONCLUSION

Jericho Systems Corporation is committed to providing our customers in the DoD, federal agencies, and commercial enterprises access control solutions that optimize open architectures, open protocols, and as appropriate, open source software.

Our subject matter experts stay current with open source use within access control solutions and can help customers address relevant supply chain issues. Our deep knowledge of access control architecture, design, development, and deployment provide value to enterprise departments with limited resources.

Our approach combines the best of commercial products and services with an open philosophy — allowing us to deliver robust, dynamic access control that simultaneously reduces supply chain risk, increases interoperability, and decreases maintenance and implementation timelines.

ABOUT JERICHO SYSTEMS

Jericho Systems Corporation provides enterprise-scalable access control, decision-making, content-filtering, privacy-enabling, and privilege management solutions, with its largest customers representing the healthcare, Department of Defense, Intelligence, and Homeland Security communities. Jericho specializes in fine-grained, policy-based data filtering technology that supports dynamic, attribute-based access control (ABAC). For additional information on Jericho Systems Corporation, please visit our website at http://www.jerichosystems.com.

[34] Total cost of ownership of open source software: a report for the UK Cabinet Office supported by OpenForum Europe, November 2011, http://eprints.lse.ac.uk/39826/